EXHIBIT E

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner,

v.

SINGULAR COMPUTING LLC,
Patent Owner.

Patent No. 9,218,156
Filing Date: March 25, 2013
Issue Date: December 22, 2015

Inventor: Joseph Bates
Title: PROCESSING WITH COMPACT ARITHMETIC PROCESSING ELEMENT

DECLARATION OF SUNIL P. KHATRI, Ph.D.

Case No. IPR2021-00165

TABLE OF CONTENTS

I, Sunil P Khatri, declare as follows:

1. I have been asked by counsel for Patent Owner Singular Computing LLC ("Singular" or "Patent Owner") to review U.S. Patent No. 9,218,156 (the "'156 Patent") entitled PROCESSING WITH COMPACT ARITHMETIC PROCESSING ELEMENT, and to provide my technical review, analysis, insights, and opinions regarding the '156 Patent in view of the prior art cited by Petitioner Google LLC ("Google" or "Petitioner"). I submit this declaration in support of Patent Owner's Response in this IPR proceeding. I have personal knowledge of the matters stated herein and would be competent to testify to them if required.

2. I have been retained by Singular for the above-captioned *inter partes* review proceeding. I understand that the '156 Patent is currently assigned to Singular Computing LLC.

3. I am over 18 years of age. I have personal knowledge of the facts stated in this Declaration and could testify competently if asked to do so.

I. INTRODUCTION

A. Background and Qualifications

4. I have reviewed and am familiar with the specification of the '156 Patent. I understand that the '156 Patent has been provided as Exhibit 1001. I will use the column and line numbers of the '156 Patent to to refer to particular portions of the specification (using the format [column no.]:[line no.]).

1) Research and Professional Experience

5. I have been serving as Professor in Electrical and Computer Engineering (ECE) at Texas A&M University, College Station, Texas, since September 2015. At Texas A&M, I have held the titles of Professor, Associate Professor (between September 2010 and September 2015), and Assistant Professor (between June 2004 and September 2010). My research is conducted in three main areas – computer systems, including computer architecture from the circuits up, algorithm acceleration using GPUs, FPGAs, and custom ICs and VLSI circuits; logic and its applications; and interdisciplinary extensions of the first two areas. A detailed description of each of these research areas is included in my *curriculum vitae* submitted concurrently herewith as **Exhibit 2044**.

6. From September 2020 to August 2021, I was on sabbatical at Arizona State University (Tempe, AZ) and the Air Force Research Laboratory (Rome, NY), continuing my past work on several topics including neuromorphic circuits, approximate computing, hardware security and hardware machine learning.

7. From September 2011 to August 2012, I was on sabbatical at the University of Texas at Austin, working with Professor Jacob Abraham on several topics such as genomics, sinusoidal signal-based data transfer and medical electronics.

8. From January 2000 to May 2004, while working as an Assistant Professor in Electrical and Computer Engineering at the University of Colorado at Boulder, I performed research on VLSI logic design automation, VLSI layout design automation, VLSI design methodologies to address Deep Submicron (DSM) issues such as cross-talk and power, along with interdisciplinary extensions. One of the key research areas that I worked on had to do with regularity in Integrated Circuit design using a "layout fabric" which I developed during my Ph.D. research.

9. From August 1993 to December 1999, I worked as a Research Assistant with the CAD group under Professors Robert Brayton and Alberto Sangiovanni-Vincentelli at the University of California at Berkeley. My research topics included CAD and DSM design, Sets of Pairs of Functions to be Distinguished (SPFDs), Binary Decision Diagrams, Engineering Change, Hierarchical Synthesis and Verification, Model Matching and Combinational Verification, Timing Analysis in the Presence of Cross-talk and Multi-valued Logic Synthesis. My Ph.D. thesis was the first work that indicated that a regular layout "fabric" should be used to alleviate DSM problems like cross-talk, manufacturing issues, delay variation, and signal integrity.

10. From August 1989 to July 1993, I worked as a Design Engineer with Motorola's MC88110 RISC and PowerPC 603 microprocessor groups in Austin,

Texas. During that time, I was involved in various design areas from Design for Testability to Digital and Analog Circuit Design and high-level design. I was also independently responsible for the design of the factory test controller for the MC88110 and became familiar with various ad-hoc and structured test methodologies. I designed digital and analog circuitry, as well as the MC88110's input/output buffers and clock PLL logic. In addition, I performed attendant tasks in a "vertical" VLSI design methodology, including high-level modeling, layout design and verification, full-chip integration, and global and detailed routing.

11. From August 1988 to July 1993, I worked as a Researcher with Professor M. Ray Mercer's group at the University of Texas at Austin, researching topics such as IC testing and Boolean function representation using Canonical XOR-based circuit decompositions, using a multi-level network of symmetric functions.

12. From August 1987 to July 1989, I worked as a Researcher at the University of Texas at Austin. My research included Computer Architecture and Memory Interface design, applied in the context of the METRIC multi-threaded RISC microprocessor which was being developed by Professors Donald Fussell and Roy Jenevein at the time.

2) Education

13. In 1987, I received a Bachelor of Science degree in Electrical Engineering from the Indian Institute of Technology in Kanpur, India. I maintained a GPA of 3.72 and was ranked fourth in my class of sixty students.

14. In 1989, I obtained my Master of Science degree from the Department of Electrical and Computer Engineering at the University of Texas in Austin, Texas. I was awarded the Microelectronics and Computer Development (MCD) Fellowship and maintained a GPA of 3.909.

15. The thesis for my Master's Degree was titled "*The Design of the METRIC Memory Interface and Memory System*" and involved research surrounding the design of the memory interface of METRIC, a multi-threaded RISC Microprocessor.

16. From 1993 to 1999, I attended the University of California at Berkeley, where I obtained my Doctorate Degree from the Department of Electrical Engineering and Computer Sciences. From 1993 to 1994, I was awarded the California MICRO Fellowship. I maintained a GPA of 3.963 during my Ph.D. studies.

17. My Dissertation was titled "Cross-talk Noise Immune VLSI Design using Regular Layout Fabrics".

3) Publications, Awards, Grants, and Summary

18. I have authored a total of over 268 peer-reviewed publications, including 42 journal papers, 184 conference papers, and 42 workshop papers. Among these papers, five have received a best paper award, while six other papers received best paper nominations. Additional journal papers and conference papers are currently undergoing peer review. In addition, I have co-authored 9 research monographs, 1 edited research monograph, and 3 book chapters.

19. I was invited to serve as a panelist at a conference seven times and have presented two conference tutorials.

20. I am currently a named inventor on six United States patents and one provisional patent application, with two applications currently pending.

21. I have four current research grants totaling $4.47M (of which my portion is $602,500). The total amount of the grants in which I have been involved in to date is $17.53 million, of which $2.85 million is my portion. Some of these grants are with colleagues in the Electrical and Computer Engineering department, as well as other academic departments at Texas A&M University. My research has been funded both through the government and through industrial sources.

22. I currently serve, or have served, as the following:

- Associate Editor, ACM Transactions on Design Automation of Electronic Systems
- Associate Editor, IEEE Transactions on Computers

- Associated Editor, MDPI Journal of Electronics
- EDA Track Co-Chair for ICECS (2014)
- Panel Chair for Texas WISE (2014)
- Track Co-Chair for ICECS (VLSI Systems, Applications and Computer Aided Design Track) (2013)
- Poster Session Chair for Texas WISE (2013)
- Advisory Committee for HotPI (2013)
- Panel Session Chair for SLiP (2013)
- Track Chair (Logic Track) for ICCAD (2009-2010 and 2015-2017)
- Track Chair (Logic Track) for DAC (2016-2017)
- General Chair for IWLS (2009)
- Technical Program Chair for IWLS (2008)
- Track Co-Chair (Computer Aided Network Design (CANDE) Track) for ISCAS (2008-2010)
- Track Co-Chair (Test and Methodologies Track) for ICCD (2007)
- Panel Chair for ITSW (2009)
- Publicity Co-Chair for GLS-VLSI (2009)
- Member of the TPC (several conferences)
- Session Chair (several conferences)

23. I have also received several awards, including the "Outstanding Professor Award" from the Electrical and Computer Engineering Department at Texas A&M University in 2007 and 2020, the "Association of Former Students' Distinguished Achievement Award in Teaching" in 2009, and the "Association of Former Students' College-level Teaching Award" in 2019.

24. I have over 34 years of research and professional experience in the field of Electrical and Computer Engineering, including in the areas of computer systems, logic synthesis and algorithms for VLSI Design and interdisciplinary extensions.

25. I have taught numerous undergraduate and graduate level courses at Texas A&M in the area of Electrical and Computer Engineering. I have graduated twelve Ph.D. students, eighteen M.S. students, and twelve B.S. Honors students. I am currently advising four Ph.D. students, one M.S. student, and two B.S. research students. I have also advised 37 undergraduates, four of which received an award for their research. Eight research papers in international conferences (one invited) have resulted from my work with undergraduates, and the dissertation of one of my Ph.D. students was nominated for the ACM Best Dissertation Award in 2014.

26. On the basis of my education and the experience described above, I am qualified to give the opinions set out herein.

II. COMPENSATION

27. My compensation for time worked on this proceeding is not dependent on the outcome of this proceeding, or the substance of my opinions. My compensation for time worked on this proceeding is at my customary rate of $600/hour. I have no financial interest in, or affiliation with, the Patent Owner or any of the real parties in interest.

III. MATERIALS CONSIDERED

28. In providing my technical review, analysis, insights, and opinions, I have considered the ’156 Patent and its prosecution history.

29. I have also considered the Petition filed by the Petitioner in this proceeding and the relevant exhibits relied on by Petitioner, including the expert declaration submitted by Richard Goodin. I have also considered the exhibits cited herein. To the extent that this declaration does not explicitly reference specific arguments, positions, opinions, or statements made by Petitioner – or by Petitioner’s expert, Mr. Goodin – this should not be construed to mean that I agree with those arguments, positions, opinions, or statements.

30. I have also considered my own experience and knowledge, as discussed above and described more fully in my CV, in the areas including VLSI circuit design, computer architecture, approximate computing, parallel computing, parallel software, and hardware-software co-design.

IV. LEGAL PRINCIPLES

31. I understand that a patent claim is unpatentable as "obvious" if the subject matter of the claim as a whole would have been obvious to a person of ordinary skill in the art (POSA) as of the time of the invention at issue.

32. I understand that the use of "the person of ordinary skill" rubric is to prevent one from improperly, in the present day, using hindsight to decide whether a claim is obvious.

33. I understand that the following factors must be evaluated to determine whether the claimed subject matter is obvious: (1) the scope and content of the prior art; (2) the difference or differences, if any, between the scope of the patent claim and the scope of the prior art; and (3) the level of ordinary skill in the art at the time of the invention.

34. I understand that certain secondary considerations, such as commercial success, skepticism of experts, unexpected results, and copying, may provide evidence of non-obviousness. I further understand that such considerations are often the most probative and determinative of obviousness or non-obviousness.

35. I understand that I must construe a claim in accordance with the ordinary and customary meaning of the language of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent.

A. Level of Skill In the Art

36. I understand that I should perform my analysis from the viewpoint of a person of ordinary skill in the art. I understand that this hypothetical person of ordinary skill in the art is considered to have the normal skills of a person in a certain technical field. I understand that factors that may be considered in determining the level of ordinary skill in the art include, *e.g.*, the types of problems encountered in the art, prior art solutions to those problems, the sophistication of the technology, and the education level of active workers in the field.

37. I agree with Petitioner's proposed level of skill in the art, except that I disagree that such a person would have more than two years of experience. Therefore a POSA would be: a person with a Bachelor's degree in Computer Science, Electrical Engineering, or Applied Mathematics, with 2 years of academic or industry experience in computer architecture. Pet. at 8-9.

B. The Claimed Invention Of The '156 Patent

38. The '156 Patent is entitled "Processing with Compact Arithmetic Processing Element" and issued on December 22, 2015. The '156 Patent claims priority, through parent and grandparent applications, to U.S. Provisional Patent Application No. 61/218,691, filed on Jun. 19, 2009. I have reviewed the '156 Patent and its file history.

39. The inventor of the '156 Patent, Dr. Bates, recognized that even though then-modern conventional microprocessors contained about one billion

transistors, they could perform only a handful of operations per clock cycle. '156 Patent, 1:55-63. Dr. Bates explained that a large portion of this inefficiency comes from using transistor-intensive full-precision arithmetic units:

> As described above, today's CPU chips make inefficient use of their transistors. …they deliver great precision, performing exact arithmetic … standardized arithmetic with 32 and 64 bit floating point numbers. Many applications need this kind of precision. As a result, conventional CPUs typically are designed to provide such precision, using on the order of a million transistors to implement the arithmetic operations.

'156 Patent, 3:11-3:26.

40. However, Dr. Bates realized that such full-precision, inefficient components were not necessary for all applications, including many valuable ones:

> There are many economically important applications, however, which are not especially sensitive to precision and that would greatly benefit, in the form of application performance per transistor, from the ability to draw upon a far greater fraction of the computing power inherent in those million transistors. Current architectures for general purpose computing fail to deliver this power.

'156 Patent, 3:27-33.

41. The '156 Patent is thus directed away from prior art computers based on full-precision execution units that take up space and are wasteful of transistors.

As Dr. Bates further explains in the specification, "[b]ecause LPHDR processing elements are relatively small, a single processor or other device may include a very large number of LPHDR processing elements, adapted to operate in parallel with each other." '156 Patent, 6:56-59. As a result, "embodiments of the present invention may be implemented as any kind of machine which uses LPHDR arithmetic processing elements to provide computing using a small amount of resources (e.g., transistors or volume) compared with traditional architectures." '156 Patent, 8:8-12.

42. By using a "very large number" of LPHDR execution units in parallel, computer systems are able to achieve significantly better performance than prior art systems. Because each LPHDR execution unit requires fewer resources (*e.g.*, fewer transistors, less physical volume) than a full-precision execution unit, "there is a large amount of arithmetic computational power per unit of resource. This enables larger problems to be solved with a given amount of resource than does traditional computer designs." '156 Patent, 23:37-44; *see also id.*, 6:56-59. In particular, the claimed systems "might perform tens of thousands of arithmetic operations per cycle, as opposed to hundreds in a conventional GPU or a handful in a conventional multicore CPU. '156 Patent, 23:46-49.

43. In addition, the '156 Patent also teaches computer systems in which the number of LPHDR execution units exceeds the number of full precision execution units:

> For certain devices … according to the present invention, the number of LPHDR arithmetic elements in the device (e.g., computer or processor or other device) exceeds the number, possibly zero, of arithmetic elements in the device which are designed to perform high dynamic range arithmetic of traditional precision (that is, floating point arithmetic with a word length of 32 or more bits).

'156 Patent, 27:52-59.

44. The increased level of compute parallelism and scale in such computer systems is necessarily achieved at the cost of precision—the vast majority of the high dynamic range floating-point operations performed by the device must be performed at low precision. Dr. Bates was the first to understand that sacrificing precision for increased parallelism/scale results in significant performance gains per unit of resource over the prior art. In fact, Dr. Bates notes that when certain applications are implemented using a device that uses LPHDR execution units, the final application error is significantly lower than the error of the LPHDR execution units themselves. '156 Patent, 16:59-23:34.

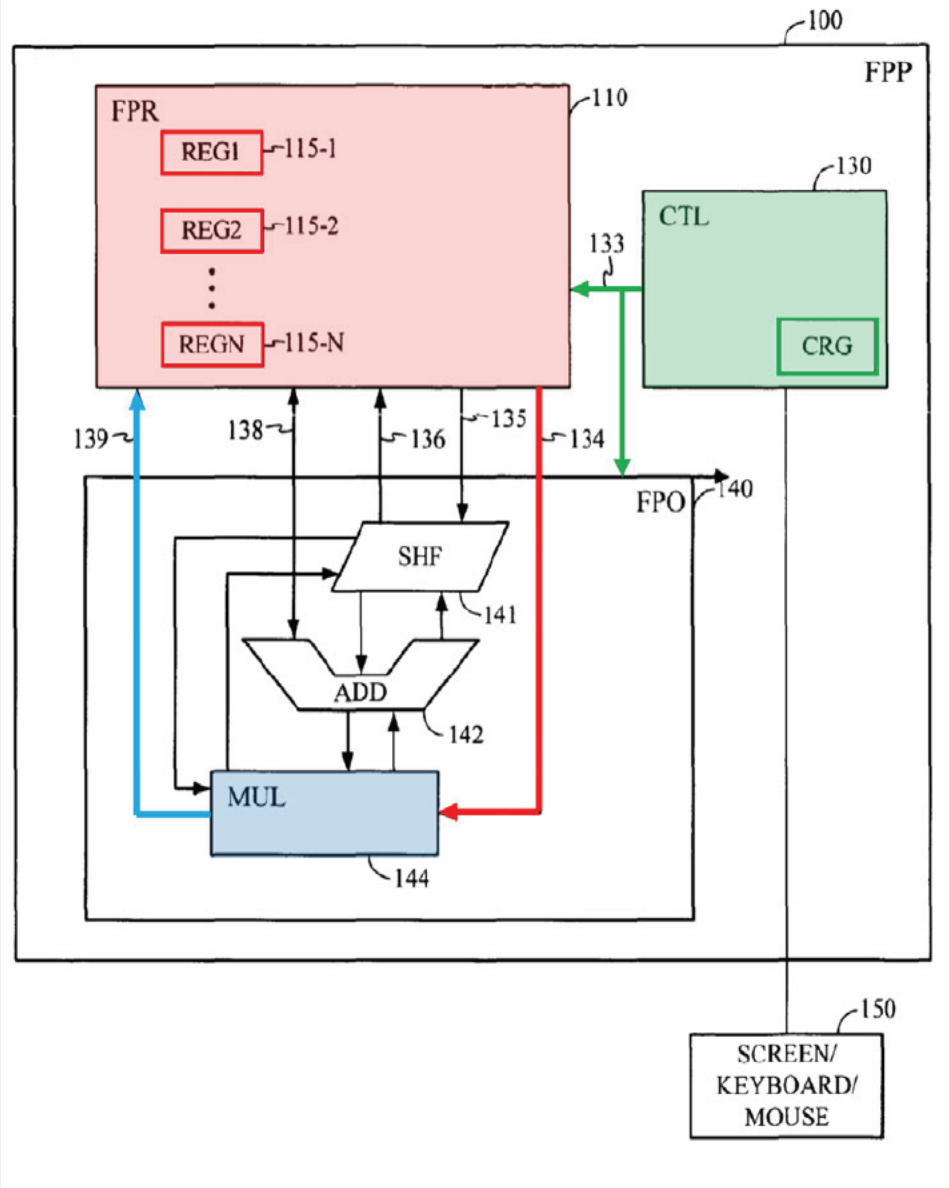
V. THE CITED REFERENCES

A. Dockser

45. I have reviewed U.S. Pat. Publ. 2007/0203967, titled "Floating-Point Processor With Reduced Power Requirements For Selectable Subprecision" ("Dockser"). Dockser discloses the use of a full-precision floating point processor (FPP) that can selectably reduce precision in order to reduce its power draw. *See, e.g.*, Ex. 1007, Dockser ¶ [0026].

46. Dockser's teaching of an execution unit that is always capable of operating in both full-precision mode (its default) and reduced-precision modes is motivated by the prevalent view in the prior art that while low-precision operation might be acceptable for "certain applications," for a general-purpose processor, full-precision capability is "needed." Dockser ¶¶ [0003, 0018], claims 1, 8, 15, 20. As such, Dockser includes both full and reduced precision modes. *See e.g.*, Dockser ¶¶ [0028], [0017], [0014], [0023], Fig. 2.

47. The components of Dockser's FPP is shown in Figure 1, reproduced below:



Dockser, Fig. 1 (Annotated)

Dockser's FPP has a register file (FPR 110, shaded in **red**) comprising registers (red boxes) that hold IEEE 32-bit full-precision values. It also includes a controller (CTL 130, shaded in **green**) with a control register (**green** box) that can store "subprecision select bits" corresponding to the desired level of precision. Dockser, Fig. 1; *see also id.*, ¶¶ [0017-18]. The FPP performs arithmetic operations (e.g.,

including addition and multiplication) at full-precision – or at the desired sub-precision using values stored in the register file as operands. *See id.*, ¶ [0019], [0017], [0028].

48. If no sub-precision is selected, Dockser's FPP performs arithmetic at "maximum precision" (*i.e.*, 32-bit IEEE full-precision). *See* Dockser, claims 1, 8, 15, 20; *see also id.*, ¶ [0018] ("the floating-point controller 130 *may* be used to select the sub-precision of the floating-point operations …")). Otherwise, the operation is performed at a reduced precision. In reduced-precision modes, Dockser's FPP reduces the mantissa size by powering-down some of the cells in the register file that store the mantissa, and also powers down parts of the arithmetic logic circuits (contained, *e.g.*, within the multiplier, shaded in **blue** above) that are not needed for the selected subprecision. *Id.*, ¶¶ [0026-27].

49. Unlike the '156 Patent, Dockser does not teach systems that include multiple arithmetic units operating in parallel. Instead, Dockser discloses a single floating-point execution unit in isolation. The absence of a parallel teaching is not surprising, given that Dockser's selectable low-precision unit is designed to conserve power in "battery operated devices where power comes at a premium, such as wireless telephones, personal digital assistants (PDA), laptops, game consoles, pagers, and cameras." Dockser, ¶ [0003]. A POSA would not understand Dockser to disclose or suggest using its FPP units in parallel.

50. Dockser discloses that it uses "conventional" arithmetic units to perform operations, such as addition and multiplication. Dockser, ¶ [0020].

51. Because Dockser's execution unit is capable of full-precision, and uses conventional arithmetic units, a POSA would understand that it would have at least as many transistors and take up at least as much space as a conventional full-precision arithmetic unit, even when operating in a reduced-precision mode. Indeed, because Dockser's execution unit includes additional control circuits for selecting reduced-precision modes, a POSA would expect it to be *larger* than a conventional full-precision execution unit, making it unsuitable for scaling, and therefore unsuitable for use in parallel processing arrays.

B. Tong

52. I have reviewed "Reducing Power by Optimizing the Necessary Precision/Range of Floating-Point Arithmetic" by Jonathan Ying Fai Tong et al. ("Tong"). Tong teaches that using lower-precision arithmetic can reduce power consumption. *See generally* Tong, pp. 273-285. Tong is based on the commonly held belief that some applications can only be performed using full-precision arithmetic: "[e]ven though we may be able to assume that most of our operands can be computed successfully in limited precision, it appears inevitable that some fraction of our operands will require full IEEE-standard precision." *Id.* (emphasis added).

53. Importantly, like Dockser, Tong is focused on reducing power consumption, and does not teach parallel processing systems that include multiple arithmetic units that operate simultaneously, let alone systems having much larger numbers of low-precision units than full-precision units. Each low-precision unit disclosed by Tong is either as large as a full-precision unit or paired with a full precision unit on a 1:1 basis (Tong, 282).

54. For example, Tong teaches "simply including both full and reduced precision FP units and using appropriate sleep-mode circuit techniques to shut down the unused unit." Tong, 282. In this approach, each low-precision unit is paired with a full-precision unit in a 1:1 ratio, requiring more physical space than a full-precision unit alone. Indeed, this approach is presented as an option in situations where silicon real-estate is not at a premium. *See* Tong, 282 ("Given the decreasing cost of silicon area …").

55. In another example, Tong also teaches a "digit-serial" multiplication circuit that, using control signals, is operable to perform a reduced-precision operation in a single clock cycle. The result of this reduced-precision operation, in which an 8-bit operand is multiplied with a 24-bit operand, can be combined with other reduced-precision results over multiple clock cycles in a process called "digit-serial multiplication," yielding a full-precision result. When performing low-precision operations, Tong's digit-serial multiplier does consume less power than a

traditional full-precision unit. However, it occupies *more* physical space than a full-precision multiplier, because it "require[s] extra random logic for control of the multiple passes through the digit serial structure." Tong, 281; *see also* 280 ("The lower precision digit-serial design is slightly larger [than the full-precision execution unit]"), Table V (showing that Tong's digit-serial multiplier occupies ~3% more area than a conventional full-precision multiplier). Therefore, a POSA would understand that like Dockser's FPP, the execution units taught by Tong are capable of reduced-precision operation, but are *larger* than a conventional full-precision execution unit. A POSA would understand that this large size would make them unsuitable for scaling, and therefore unsuitable for use in parallel processing arrays.

56. A POSA would therefore understand that as with Dockser, Tong is focused on power-savings, does not even mention increasing computational scale and parallelism, and certainly does not teach a computer comprised of a much larger number of low-precision units compared with full-precision units.

C. MacMillan

57. I have reviewed U.S. Patent No. 5,689,677, titled "Circuit For Enhancing Performance Of A Computer For Personal Use" ("MacMillan"). MacMillan is directed to a computer system that includes a host processor and "a plurality of processing elements." Macmillan, 12:39. MacMillan does not describe

the capabilities of each "processing element" (PE) in detail, noting only briefly that "[i]nteger and floating point accelerators could be included in each PE." MacMillan, 12:55-56.

58. Regarding arithmetic precision/imprecision, MacMillan is silent. It simply teaches that each PE can "perform atomic operations on data values up to *32 bits wide*." MacMillan, 7:8-9. A POSA would understand that unlike Dockser and Tong, MacMillan is not focused on minimizing power consumption, and is not specifically tailored for use in portable or mobile devices. Indeed, MacMillan's only reference to power consumption relates to heat dissipation inside the "cabinet" of a workstation. *See* MacMillan, 3:4-6.

VI. CLAIM CONSTRUCTION

A. "Low Precision High Dynamic Range (LPHDR) Execution Unit"

59. In my opinion, the Board should construe the term "low precision high dynamic range (LPHDR) execution unit" as "an execution unit that executes arithmetic operations only at low precision and with high dynamic range, wherein 'high dynamic range' and 'low precision' are defined according to the numerical requirements below."[1] A POSA would understand this as the plain meaning of the term as read in light of the specification.

[1] Where "below" refers to the remainder of the claim in which the term appears (*e.g.*, the numerical requirements that specify a minimum level of imprecision and a minimum dynamic range).

60. A POSA would read the term "low precision high dynamic range (LPHDR) execution unit" to limit the execution unit as "low precision" and "high dynamic range." This reading necessarily excludes full-precision or mixed full and low precision units. This comports with the plain and ordinary meaning of the term, and basic engineering knowledge—at the most basic level, a "low precision" execution unit cannot be a "high precision" execution unit. Other limitations of the claims—" wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/65,000 through 65,000" and "for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input" would inform a POSA of exactly what the boundaries of "high dynamic range" and "low precision" are.

61. An interpretation that any execution unit that "meets each claim's recited error amount" is a "low precision" execution unit—or that otherwise allows the execution unit to act with high precision—would effectively read the term "low precision" out of the claims entirely. That is, the term "at least one first low

precision high dynamic range (LPHDR) execution unit adapted to execute" would have exactly the same scope as "at least one execution unit adapted to execute." A POSA would not ignore a key claim limitation in that fashion.

62. The proposed construction is supported in the specification. The specification explains that the "LPHDR execution units" of the '156 patent are based on a "fundamentally different approach" from the units that incorporate full-precision capabilities as taught in the prior art. '156 Patent, 6:1-5. These prior-art units are described in 156 5:36-67. Unlike the prior art full-precision units, the claimed LPHDR execution units are smaller and have fewer transistors:

> "For example, embodiments of the present invention may be implemented as any kind of machine which uses LPHDR arithmetic processing elements to provide computing *using a small amount of resources* (*e.g.*, transistors or volume) ***compared*** *with traditional architectures*."
>
> '156 Patent, 8:8-16 (emphasis added)

63. LPHDR execution units utilize less chip real-estate than conventional execution units *precisely because* they do not include full-precision arithmetic circuits:

> "One variety of LPHDR arithmetic represents values from one millionth up to one million with a precision of about 0.1%.... One example of an alternative embodiment is to use a logarithmic representation of the values.. *As a result*, *the area* of the arithmetic circuits *remains* ***relatively*** *small* and a *greater number* of computing elements *can be fit into a given area* of silicon … which gives it an advantage for those computations able to be expressed in the *LPHDR* framework."

'156 Patent, 6:8-27 (emphasis added)

64. The small size and low transistor count of each individual LPHDR execution unit is what allows a much larger number of LPHDR execution units to operate in parallel on a single chip:

> Because *LPHDR processing elements are relatively small,* a single processor or other device may include a very large number of LPHDR processing elements, adapted to operate in parallel with each other, and therefore may constitute a massively parallel LPHDR processor or other device.

'156 Patent, 6:56-60 (emphasis added). The specification contrasts the small LPHDR units with units, like GPUs, that can operate in both high and low precision. *Id.*, 5:36-45.

65. As the specification explicitly states, the fact that "LPHDR execution units" are smaller than full-precision units is not limited to a preferred embodiment, but is an essential aspect of the invention as a whole:

> The discovery that massive amounts of LPHDR arithmetic is useful as a fairly general computing framework, as opposed to the common belief that it is not useful, can be an advantage in any (massively or non-massively) parallel machine design or non-parallel design, not just in SIMD embodiments. It could be used in FPGAs, FPAAs, GPU/SIMT machines, MIMD machines, and *in any kind of machine that uses* ***compact*** *arithmetic processing elements* to perform large amounts of computation using a small amount of resources (like transistors or volume).

'156 Patent, 24:8-17; *see also id*, 25:23-29; Title ("Processing with *Compact* Arithmetic Processing Element").

VII. THE CHALLENGED CLAIMS ARE VALID

A. Ground 1: Claims 1-2, and 16 Are Not Obvious Over Dockser

66. Dockser does not render obvious any challenged claim of the '156 Patent because each claim of the '156 Patent requires a ***low precision*** high dynamic range (LPHDR) execution unit. Dockser neither discloses nor renders obvious this limitation.

1) Dockser Is Not a Low Precision High Dynamic Range (LPHDR) Execution Unit

67. As discussed above, the proper construction of "low precision high dynamic range (LPHDR) execution unit" is "an execution unit that executes arithmetic operations only at low precision and with high dynamic range, wherein 'high dynamic range' and 'low precision' are defined according to the numerical requirements below."

68. Even if the Board chooses to apply the plain meaning of the term "low precision," a POSA would understand that an LPHDR execution unit is necessarily "low precision," and therefore not capable of high or full precision. As I state above, a POSA would not read "low precision" out of the term "LPHDR execution unit."

69. Dockser, by contrast, is capable of full-precision operation, and is therefore relatively large and "wasteful of transistors" (*See supra* citing '156 Patent

5:36-45). As I state above, a POSA would understand that Dockser's 32-bit FPP includes all of the circuitry needed for full precision arithmetic on data in IEEE 32-bit format, and *also* has additional circuitry and transistors, such as the controller and associated logic, to allow selection of sub-precisions. Dockser, ¶¶ [0019]-[0020], [0026-27]. This additional circuitry is required for the FPP to perform operations such as multiplication with selectively reduced precision, while retaining its ability to perform those same operations with full precision. *See* Dockser, ¶ [0026] ("The subprecision select bits *may be* used to reduce the precision of the floating-point operation." (emphasis added)); *see also id.* ¶ [0028] (referring to "the full precision mode). Indeed, Dockser explicitly anticipates scenarios in which this subprecision capability will *not* be used, because for certain applications "a greater precision may be needed." *Id.*, ¶ [0003]. Dockser does not disclose any embodiments of an execution unit without either full-precision or the additional circuitry / transistors for selecting a subprecision. A POSA would understand that Dockser's large size makes it unsuitable for increasing the scale of computation which is needed for parallelism as taught in the '156 patent.

70. Dockser's entire stated purpose is to provide a processor that can perform operations at various precisions, from full precision to lower precision. Dockser, ¶ [0003] ("For general purpose processors, however, the common situation is that for certain applications, e.g. generating 3D graphics, a reduced

precision may be acceptable, and for other applications, e.g. implementing Global Positioning System (GPS) functions, a greater precision may be needed. As a result, there is a need in the art for a floating-point processor in which the reduced precision, or subprecision, of the floating-point format is selectable."); Dockser, ¶ [0028] ("The floating-point addition operation in the ***full precision mode*** is performed through a succession of stages…" (emphasis added)). Dockser further explains that "the precision for one or more floating-point operations *may* be reduced from that of the specified format." Dockser ¶ [0014] (emphasis added). A POSA would be aware of the foregoing excerpts, including Dockser's focus on power-saving and maintaining full precision capability, and would thus understand that Dockser is a full-precision 32-bit FPP, capable of operating at lower precisions to save power. This would disqualify it from being considered as an LPHDR execution unit,

71. Further, Dockser's FPP has the ability to perform full-precision 32-bit floating-point computations as well as lower precision floating-point computations, thereby rendering it too big and too wasteful of transistors to increase the scale of computation in a parallel LPHDR framework as taught by the 156 patent. This would disqualify it from being considered an LPHDR execution unit.

72. The understanding of a POSA would be particularly informed by the '156 Patent's description of a variable-precision GPU as background art.

Specifically, the specification mentions a GPU which comprises multiple execution units that support both half-precision ("16 bit floating point") for "those applications that want it", and full-precision ("32 bit floating point" or "64 bit floating point") operations "because they are believed to be needed for traditional graphics applications". These GPUs are characterized as devices "[that] devote substantial resources to 32 … bit arithmetic and are wasteful of transistors." A POSA would understand this to teach that the '156 patent explicitly distinguishes a variable-precision floating point execution unit in a GPU (which is "wasteful of transistors") from an LPHDR execution unit (which uses transistors efficiently, as I explain above). Furthermore, Petitioner does not argue or show that it would have been obvious to modify Dockser's full-precision FPP to become such an LPHDR execution unit.

73. Therefore, in my opinion, Dockser does not render obvious any of the challenged claims.

B. Ground 2: Claims 1-2, 16, and 33 Are Not Obvious Over Dockser in View of Tong

1) The Combination of Dockser and Tong Does Not Disclose or Render Obvious any Challenged Claim

74. I understand that Petitioner does not argue that Tong alone discloses or renders obvious an LPHDR execution unit. *See* Pet., 40-47. As I state above, Dockser does not disclose or render obvious an LPHDR execution unit. Since

Tong does not remedy that deficiency, in my opinion the combination of Dockser and Tong does not render any claim obvious.

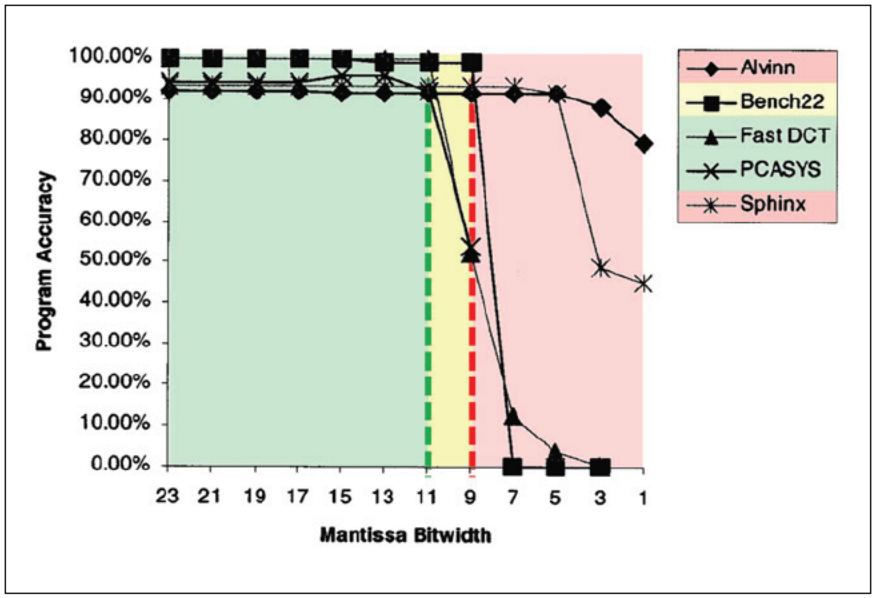
75. In particular, Petitioner argues that the combination of Dockser and Tong renders claims 1-2, 16, and 33 obvious because it "would have motivated a POSA to configure Dockser's FPP to operate at the precision levels Tong teaches." Pet., 43.

76. However, even so configured, a Dockser FPP is not an "LPHDR execution unit" under the construction proposed above, because it would still be adapted to perform full-precision arithmetic operations. As I explain above, a person of ordinary skill in the art would understand that an LPHDR execution unit is one that executes arithmetic operations only at low precision and with high dynamic range. By contrast, regardless of the way it is configured at any particular time, a Dockser FPP retains the capability of performing full-precision operations, and can be programmed to do so by simply changing the value of the "subprecision select bits," as I explain above. For this reason, programming a Dockser FPP to operate at a particular subprecision does not turn it into an "LPHDR execution unit."

2) Petitioner's Claim that Tong Teaches Using only 5 Mantissa Bits Is Incorrect, which Means a Dockser and Tong Combination Lacks the Imprecision Required to Meet the '156 Patent's X/Y imprecision limitations

77. Petitioner incorrectly argues that Tong teaches an "optimum precision" of "between 5 and 11 mantissa fraction bits." Pet., 57-58. In fact, a POSA would understand that Tong teaches that *at least 11* mantissa bits are required for consistent performance, even for a limited, experimental "set of five signal processing applications" which are "programs dealing with human interfaces [that] process sensory data with intrinsically low resolutions." Tong, 278. Tong further demonstrates that fewer than 11 bits of mantissa result in the failure of a significant fraction of test applications. *Id.* Further, neither Petitioner nor Petitioner's expert, Mr. Goodin, has provided any evidence that arithmetic with an 11-bit mantissa meets the X/Y limitations required for LPHDR operations in the claims of the '156 patent.

78. A POSA would understand that Tong's experimental results show that using fewer than 11 mantissa bits unacceptably reduces accuracy for the benchmark applications *that were specifically selected* (from a category of programs) for their high tolerance for imprecision:



Tong, Fig. 6 (Annotated).

79. These experimental results are shown above, with color annotations denoting the precision levels suitable for general-purpose operation (**green**), those at which a significant percentage of applications (60%) begin to produce unacceptable results (**yellow**), and finally, levels of precision that are unsuitable for most testing benchmarks (**red**). At 5-bit precision, only 2 out of 5 benchmark applications produce acceptable accuracy.

80. Accordingly, in my opinion, Tong does not teach, and would not motivate a POSA, to use five mantissa bits with Dockser or any other implementation.

3) There Is No Motivation to Combine Dockser and Tong

81. A POSA would not have been motivated to combine Dockser and Tong. Tong and Dockser both disclose variable precision processors; because of

this overlapping functionality, Tong does not disclose any extra capability that would have motivated a POSA to use it with Dockser.

82. Further, a POSA would not have been motivated by Tong to use 5 bits of precision, because Tong teaches that with 5 bits of precision, only 2 out of 5 example applications provide sufficient accuracy (where Tong's example applications, as I explain above, are selected from a class of applications that use "intrinsically low precision" data). Tong, 278.

C. Ground 3: Claims 1-8 and 16 Are Not Obvious Over Dockser in View of MacMillan

1) Dockser's FPP is not an "LPHDR execution unit"

83. As I explain above, Dockser's FPP is not a "low precision high dynamic range (LPHDR) execution unit" as recited in all challenged independent claims under Ground 3, because it does not execute arithmetic operations "*only* at low precision and with high dynamic range." MacMillan does not remedy this deficiency.

2) Dockser and MacMillan Fail to Disclose "wherein the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide"

84. Even if the teachings of Dockser and MacMillan were combined to produce a device with, for example, 256 Dockser units operating in parallel, such a device would fail to meet the "exceeds" limitation of claims 3-8.

85. The "exceeds" limitation requires that the "number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide."[2] As I explain further below, a POSA would understand that Dockser's FPP is adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide.

86. Even if Dockser's FPP were an LPHDR execution unit (which it is not, in my opinion) the Dockser/MacMillan combination cannot meet claim 3's "exceeds" limitation, because those same "Dockser LPHDR execution units" would also be "adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide," meaning that the number of LPHDR execution units will never exceed the number of claim 3 full-precision floating point multiplication execution units. That is, even under Petitioner's incorrect argument that each Dockser FPP qualifies as an LPHDR execution unit, it *also* qualifies as a claim 3 full-precision floating-point multiplication execution unit. Therefore, no matter how many Dockser FPP's are included in a device, the number of LPHDR execution units in the device will *at most equal* to the number

[2] For clarity, I refer to "execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide" as "claim 3 full-precision floating point multiplication execution units."

of claim 3 full-precision floating-point multiplication execution units – and will never "exceed [it] by at least one hundred," as claim 3 requires.

3) Dockser/MacMillan Fails to Disclose or Render Obvious "at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit"

87. The claims require "at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit." Claim 2 further requires that that the computing device "comprises at least one of a central processing unit (CPU), a graphics processing unit (GPU), a field programmable gate array (FPGA), a microcode-based processor, a hardware sequencer, and a state machine." I understand that Petitioner identifies MacMillan's "Host CPU" as the "computing device adapted to control the operation of the at least one first LPHDR execution unit." Pet., 51-52. I disagree.

88. A POSA would understand that MacMillan makes clear that the SIMD Controller, not the Host CPU, controls the operation of the PEs. MacMillan, 13:35-36 ("Since the PEs operate under the control of the SIMD Controller…"); 8:1-10 (explaining that SIMD Controller "contains the program counter used to step through the list of instructions that may include instructions to be executed by the PEs"); 10:65-11:5 ("The SIMD Controller 252 then executes the SIMD program."); 10:3-16; Indeed, a POSA would understand that while the PEs are executing, the SIMD Controller has taken control of the host bus, and the Host

CPU has relinquished the bus. At this point, the SIMD Controller executes the SIMD program, broadcasting data and opcodes to the SIMD RAMs (which contain the SIMD PEs). When the SIMD program is complete, the SIMD Controller relinquishes the bus, and the Host CPU regains control of the bus, resulting in the SIMD Controller going idle. Thus there is no way for the Host CPU to control the PEs. *See* MacMillan, 10:4-16; 10:53-11:5; 12:35-46; *see also* Goodin Dep. 126:3-129:6. Not only is the Host CPU not disclosed as controlling the PEs, it is merely optional. *Id.* at 116:6-18.

89. In my opinion, because MacMillan's Host CPU does not control the PEs, the Dockser/MacMillan combination does not render obvious any claims.

4) Dockser's FPP Is a Claim 3 EU Because It Is Adapted to Execute at Least the Operation of Multiplication on Floating Point Numbers That Are at Least 32 Bits Wide

90. Dockser is an execution unit. I understand that Petitioner agrees with me. *See* Pet., 13.

91. A POSA would understand that Dockser is adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide under the plain meaning of the term.[3]

[3] I understand that Petitioner and Patent Owner agree that "adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide" means that the execution unit performs multiplication in "traditional"—*i.e.*, full—precision. Pet., 51-52.

92. Dockser discloses that its FPP takes input operands that are 32 bits wide and stores them in IEEE-754 format. *See e.g.*, Dockser, ¶ [0017] ("Each register location 200 is configured to store a 32-bit binary floating-point number, in an IEEE-754 32-bit single format."). Dockser also discloses that its FPP includes "a floating-point multiplier (MUL) 144 configured to execute floating-point multiply instructions." Dockser, ¶ [0019]. A POSA would therefore understand that Dockser multiplies 32-bit operands when operating in full precision mode, and therefore it is "adapted to" execute the operation of multiplication on numbers at least 32 bits wide.

93. Therefore, even if Dockser's FPP is an "LPHDR execution unit" (as Petitioner incorrectly argues), the Dockser/MacMillan device would also satisfy the "adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide" requirement, and would thus fail to satisfy the "exceeds" limitation. If, for the sake of argument, we count each Dockser FPP as an "LPHDR execution unit," then in one example, the Dockser/MacMillan device would have 256 LPHDR execution units. However, because each Dockser FPP is also "adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide," the Dockser/MacMillan device would therefore also have at least 256 execution units "adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits

wide." As a result, in my opinion, the Dockser/MacMillan cannot meet the "exceeds" limitation.

5) Petitioner's Interpretation of "adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide" Is Inconsistent with the Specification

94. I understand that Petitioner argues that Dockser's FPP does not meet the "exceeds" limitation by arguing that units are claim 3 full-precision floating point multiplication execution units ***only if*** they are "'traditional precision' execution units that do not 'sometimes' produce results different from the correct traditional-precision result." Pet., 52. I disagree.

95. A POSA would not understand "adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide" to exclude units, like Dockser, which selectably operate at full precision, as Petitioner claims. Pet., 52.

96. Instead, a POSA would understand this term to include any or all units in the device that are designed to perform "multiplication on floating point numbers that are at least 32 bits wide," even if those units can perform other operations as well. Petitioner's (and Mr. Goodin's) interpretation directly contradicts the claim language itself, which encompasses every execution unit adapted to perform "*at least"* the operation of 32-bit multiplication. If such an

execution unit performed additional operations too, it would still meet the claim language.

97. Second, Petitioner's interpretation contradicts the teachings of the specification. For example, the specification discloses an execution unit that can be both a claim 3 full-precision floating point multiplication execution unit and *a* lower precision unit (which is not an LPHDR execution unit as explained below) that 'sometimes' produces results different from the correct traditional-precision result:

> "When a graphics processor includes support for 16 bit floating point, that support is alongside support for 32 bit floating point … That is, the 16 bit floating point format is supported for those applications that want it, but the higher precision formats also are supported because they are believed to be needed for traditional graphics applications and also for so called "general purpose" GPU applications. Thus, *existing GPUs devote substantial resources to 32 … bit arithmetic and are wasteful of transistors* …"
>
> '156 Patent, 5:36-45 (emphasis added)

98. A POSA would understand that the "graphics processor" described in the passage above includes multiple execution units operating in parallel, each of which can be configured to perform both full-precision ("32 bit floating point") and half-precision ("16 bit floating point") operations. Such execution units are *both* "adapted to execute at least the operation of multiplication on floating point

numbers that are at least 32 bits wide" *and* capable of performing "operations that 'sometimes' produce results different from the correct traditional-precision result" (*i.e.*, half-precision operations that produce results in the "16 bit floating point" format).[4]

99. A POSA would understand the above passage to teach that these execution units are "wasteful of transistors" because, like the execution units of Dockser and Tong, they retain the *capability* of performing 32-bit multiplication, even if they can be configured to "sometimes" operate at lower precisions. A POSA would understand that such conventional execution units having full-precision *capability* (whether or not they execute other precision operations) are precisely what the "adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide" limitation of claim 3 is intended to capture. More importantly, a POSA would understand that this is precisely what distinguishes these conventional execution units from "LPHDR execution units" that perform arithmetic operations *only* at low precision (and thus use "a small amount of resources") in claim 3. '156 Patent, 8:9-12.

100. More generally, a POSA would understand that the specification describes claim 3 full-precision floating point multiplication execution units only by reference to their "high dynamic range arithmetic of traditional precision". *See*

[4] Not to be confused with the bfloat16 (or "brain float") format used by Google's products.

'156 Patent at 28:9-16; *see also id.* at 28:9-67. The specification does not preclude a claim 3 full-precision floating point multiplication execution unit from being able to produce incorrect results. Petitioner imports that requirement from a completely different section of the patent, taken completely out of context. *See* Pet., 52 (citing '156 Patent, 26:39-49); *see also* Goodin, ¶ 363. A POSA would understand that that section uses "sometimes" to describes the imprecision limitation of an LPHDR execution unit, as expressed in the X% limitation of the Challenged Claims. That section says nothing about claim 3 full-precision floating point multiplication execution units. *See* '156 Patent, 26:39-49.

D. A POSA Would Not Combine Dockser with MacMillan

101. A POSA would not combine Dockser with MacMillan because it would not be operable for its intended purpose.

102. Dockser is focused on the objective of reducing power consumption and is not in any way concerned with the objective of use in parallel processing arrays, while MacMillan is focused on a parallel architecture that increases computational capability, and not focused on reducing power consumption (indeed, MacMillan's only reference to power consumption relates to heat dissipation inside the "cabinet" of a workstation). *See* MacMillan, 3:4-6.

103. Incorporating Dockser's FPPs into MacMillan would defeat MacMillan's stated objective of achieving a highly parallel SIMD computer

architecture at "lower system cost." *See* MacMillan, 5:58-59 ("The invention of this shared memory results in lower system cost"). As explained above, Dockser's FPPs are even *larger* than traditional full-precision execution units because of the control circuitry needed to implement the selectable subprecision modes.

104. As a result, replacing the full-precision execution units of MacMillan with Dockser FPP units would require additional circuitry and chip space and would therefore *increase* costs, while providing no benefit. Combining Dockser and MacMillan would thus sacrifice the benefits of *both* references, which a POSA would not be motivated to do.

E. Ground 4: Claims 1-8, 16, and 33 Are Not Obvious Over Dockser in View of Tong and MacMillan

105. As I explained above with respect to Grounds 2 and 3, the combination of Dockser, Tong, and MacMillan does not disclose or render obvious an LPHDR execution unit. Similarly, as set forth above with respect to Ground 3, the combination does not disclose or render obvious the "at least one first computing device" limitation. Finally, as set forth above with respect to Ground 2, Tong does not teach using only 5 bits of mantissa. I understand that Petitioner's Ground 4 arguments do not address these deficiencies.

1) The Dockser/MacMillan System as Modified by Tong Would Not Meet the "Exceeds" Limitation

106. Even under Petitioner's incorrect construction of "LPHDR execution unit," in my opinion, the combination of Dockser, MacMillan, and Tong would still fail to satisfy the "exceeds" limitation. As I explained above, each Dockser FPP, even when operating at 5-bit precision, is an "execution unit[] adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide." Thus, even under Petitioner's incorrect construction of "LPHDR execution unit", a device with 256 (or any number) Dockser FPPs operating at 5-bit precision would have *at least* as many claim 3 full-precision floating point multiplication execution units claim 3 as "LPHDR execution units."

2) A POSA Would Not Combine Tong, Dockser, and MacMillan

107. A POSA would not be motivated to combine the teachings of Dockser, MacMillan, and Tong. As I explained above, a person of ordinary skill in the art would not have been motivated to incorporate Dockser's FPP into MacMillan, nor to combine Dockser with Tong. A POSA would understand that Tong supplies no additional motivation to combine the teachings of Dockser and MacMillan.

3) Petitioner's "Alternative" argument regarding claims 3-8 fails

108. I understand that Petitioner argues that it would be obvious to completely redesign Dockser's FPP to remove all support for its basic and default full precision mode. Specifically, Petitioner states that it would be obvious to

shrink Dockser's 32-bit register file to some unspecified lesser bit-width, and further perform some unspecified modification to Dockser's multiplier logic "to have only as many logic elements as needed to multiply mantissas of the reduced bitwidth." Pet., 58. Petitioner further states that Tong suggests a precision of "between 5 and 11 mantissa fraction bits" for "five signal processing applications." Pet., 57-8. And because two of the many applications of an embedded system disclosed in MacMillan are "signal processing" and "voice recognition," Petitioner states that a POSA would use "Dockser's FPPs in MacMillan's architecture with Tong's precision levels." Pet., 58. I disagree. This is nothing but a hindsight-based modification of the references to meet the claims of the '156 patent.

109. A POSA would not have been motivated to re-engineer the FPP of Dockser to remove its 32-bit capability based on one data-point in Tong, and then incorporate the re-engineered Dockser unit into the system of MacMillan because both MacMillan and Tong mention "signal processing." As I explain above, the entire purpose of Dockser is to provide a processor with ***selectable subprecision***. Limiting Dockser to one precision goes against Dockser's entire inventive concept, and a POSA would not be motivated to do so.

4) A POSA would not be motivated to combine Dockser, MacMillan, and Tong under Petitioner's "alternative" argument

110. A POSA would not have been motivated by Tong to remove the full-precision capabilities of Dockser and incorporate this modified Dockser FPP into the systems taught by MacMillan. Petitioner's argument is based on a fragment of a sentence in Tong taken out of context, insinuating that Tong teaches that full-precision operation "is not essential" to the functioning of a Tong or Dockser system. Pet., 58. In my opinion, this interpretation of Tong is not correct.

111. In fact, a POSA would understand that Tong teaches the opposite. Tong admits that there are "scientific programs" that "require a huge amount of precision" (Tong, 279), and broadly teaches that "it is *inevitable* that some fraction of our operands will require full IEEE-standard precision." Tong, 280 (emphasis added). Instead, like Dockser, Tong teaches systems that always have *both* full- and reduced-precision capabilities. *See*, *e.g.*, Tong, 282 (even when describing a device that has reduced precision units, it describes that system as "including *both* full and reduced precision FP units, and using appropriate sleep-mode circuit techniques to shut down the unused unit."). With this teaching in hand, a POSA would not cleave off parts of Dockser's 32-bit register file to achieve some unspecified lesser bit-width smaller execution unit, without also importing the Tong "full precision FP.". This is a convoluted, hindsight-driven modification to the fundamental teachings of the cited references.

112. In view of the above, it is clear that a person of ordinary skill would not be motivated by Tong to remove the full-precision capabilities of the Dockser units. On the contrary, Tong would only have *reinforced* the teaching of Dockser that, while reduced-precision might be a viable option in certain limited circumstances, an execution unit should retain the ability to operate at full-precision, as I explained below. See, *e.g.*, Dockser, ¶ [0003].

113. Dockser's objectives are fundamentally directed away from Petitioner's proposed "alternative" combination. As discussed above, Dockser is devoted solely to a general processor with selectable precision.

114. Nowhere does Dockser teach, or even suggest, removing its full-precision capabilities. As I explain above, a Dockser FPP is described as always needing to support a range of selectable precisions including full precision. Dockser, ¶ [0003]. Adjusting Dockser by removing its full-precision capacity violates a central tenet of Dockser—to always be able to execute full-precision operations. Dockser, ¶ [0003]. A POSA would not remove Dockser's inventive concept—supporting a range of selectable precisions including full precision. Thus, excising the full-precision circuitry of Dockser runs contrary to the main teaching and inventive feature of Dockser (as well as that of Tong, as I explain above).

115. The removal of full-precision circuitry from the Dockser FPP also goes against the teachings of MacMillan, which warns that "[t]o meet the cost objectives, the SIMD capabilities ***should not add significant complexity*** to the architecture of a computer system for personal use." *Id.*, 5:42-44. A POSA would understand that Petitioner's proposed hindsight-driven combination, which requires special, customized registers, logic elements, arithmetic units, and programming models (see below), would increase manufacturing costs and goes directly against the teachings of MacMillan, which relies on operating with conventional components to reduce cost. *E.g.*, *id.,* 6:24-26, 34-36. Further, as I explain above, the Dockser FPP, due to its control circuitry, would likely require *more* transistors than a conventional full-precision execution unit, further increasing costs. Also, the modified Dockser FPP would result in modifications to the programs that are run on the MacMillan system, which MacMillan seeks to avoid. *See* MacMillan 2:15-16 (where avoiding "the need for reprogramming" is emphasized).

5) Google Has Failed to Show That the "Alternative" Combination Would Meet the Imprecision Requirements of Claims 3-8

116. I understand that Petitioner has not shown that the resulting combination would meet the imprecision limitations. As discussed above, Tong suggests 11 bits of precision for specific signal processing applications, and

Petitioner provides no analysis of whether 11 bits of precision would meet the imprecision limitation.

6) A POSA Would Not Have Recognized the Utility of Petitioner's "Alternative" Combination of Dockser, MacMillan, and Tong

117. Finally, a POSA would not have recognized the utility of Petitioner's "alternative" combination of Dockser, MacMillan, and Tong because the combined prior art references do not teach or suggest that it would be possible to write programs that run efficiently on a Dockser/MacMillan/Tong device.

118. As the '156 patent explains, "programmers have come to think in terms of high precision and to develop algorithms based on the assumption that computer processors provide such precision …" '156 Patent, 5:63-67. The notion that low-precision computers can be programmed in the same way as general-purpose computers "is not obvious, and in fact has been viewed as clearly false by those having ordinary skill in the art." *Id*., 7:31-35. Because of this, and based on my experience and knowledge, a POSA at the time of the invention (with a Bachelor's degree and only two years of experience) would not have known how to program in low precision.

119. I understand that Google itself was concerned, even after the time of the invention, that programmers lacked the knowledge to make use of the invention. Ex. 2028 ("I think Bates is correct that there would be a lot of low-

power parallelism and very fast effective operation if … *the programming aspects alone don't kill the whole thing* ...); *id.* ("I certainly agree with you that I disagree with Joe that programmers will learn to program differently any time soon."). This further confirms my opinion that programmers at the time of the invention would not have been able to program effectively with low precision.

120. However, the '156 Patent explains that "in fact a variety of useful and important algorithms can be made to function adequately at much lower than 32 bit precision in a massively parallel computing framework, and certain embodiments of the present invention support such algorithms, thereby offering much more efficient use of transistors, and thereby provide improved speed, power, and/or cost, compared to conventional computers." '156 Patent, 7:37-44. For example, the '156 patent teaches that the "Kahan method" can be used to reduce the accumulation of errors when summing multiple low-precision results. '156 Patent, 21:65-22:14. The '156 Patent also explains how the additional computational capability of the claimed invention can be efficiently used by processing data in a "pipelined" fashion. *See* '156 Patent, 20:57 to 21:15. The specification also describes in detail several example programs that use unconventional techniques to perform various computational tasks correctly and efficiently on the claimed low-precision systems. *See generally* '156 Patent, 17:30-23:34.

121. As I have stated supra, the modified Dockser FPP would result in modifications to the programs that are run on the MacMillan system, which MacMillan teaches against. See MacMillan 2:15-16 (where avoiding "the need for reprogramming" is emphasized).

122. Without these teachings of the '156 patent, a POSA would not have been motivated to combine MacMillan and Dockser with Tong according to Petitioner's "alternative" argument because it would not have been clear to a POSA that it would be possible to program such a device to execute operations efficiently and without accumulating errors.

VIII. OBJECTIVE INDICIA OF NON-OBVIOUSNESS

123. I have reviewed [redacted]

124. It is my opinion that the experimental results obtained by Dr. Bates using his invention [redacted], would have been unexpected to those of skill in the art at the time. When the invention was filed, it was not known that a massively parallel computer that includes almost exclusively low-precision execution units, (and very

few full-precision units) would have been able to achieve high performance across a broad spectrum of applications, [redacted]

125. The fact that Dr. Bates's results were unexpected is further supported by Dockser and Tong. For example, as I explain above, Dockser teaches that for "general purpose processors, however, the common situation is that … a greater precision may be needed." Dockser, [0003]. Similarly, Tong notes that even though limited precision can be useful, "it appears inevitable that some fraction of our operands will require full IEEE-standard precision." Tong, 280. Based on the Dockser and Tong references cited by Petitioner in this proceeding, which teach that an execution unit without full-precision capability is not "generally useful," the results presented to Google by Dr. Bates would have been unexpected to a person of ordinary skill.

126. Further, based on Google's publications about the design and performance of its own TPUv2 and TPUv3 products, I conclude not only that these products are covered by some of the challenged claims (*see below*), but also that the claimed invention is central to their design and operation:

> As the size of an FP multiplier scales with the square of the mantissa width, the bf16 multiplier is half the size and energy of a fp16

> multiplier: $8^2 / 11^2 \approx 0.5$ (accounting for the implicit leading mantissa bit). Bf16 delivers a rare combination: reducing hardware and energy while simplifying software by making loss scaling unnecessary.

Ex. 2016, 7.

> As it turns out, machine learning computations used in deep learning models care more about dynamic range than they do about precision. Furthermore, one major area & power cost of multiplier circuits for a floating point format with M mantissa bits is the (M+1) ⨉ (M+1) array of full adders that are needed for multiplying together the mantissa portions of the two input numbers. The IEEE fp32, IEEE fp16 andbfloat16 formats need 576 full adders, 121 full adders, and 64 full adders, respectively. Because multipliers for the bfloat16 format require so much less circuitry, it is possible to put more multipliers in the same chip area and power budget, thereby meaning that ML accelerators employing this format can have higher flops/sec and flops/Watt, all other things being equal.

Ex. 2011, 8-9.

127. These passages, which are excerpted from documents about the design of the TPUv2 and TPUv3, show that these products both use a large number of low-precision, high dynamic range execution units (that are capable of executing arithmetic operations *only* at low precision, as I demonstrate below) and have relatively few full-precision execution units, in order to "put more multipliers in the same chip area." *Id.* This is precisely the insight Dr. Bates described and claimed in his patents and disclosed to Google in a number of presentations between 2010 and 2017.

IX. GOOGLE'S TPUV2 AND TPUV3 ARE COEXTENSIVE WITH THE CHALLENGED CLAIMS

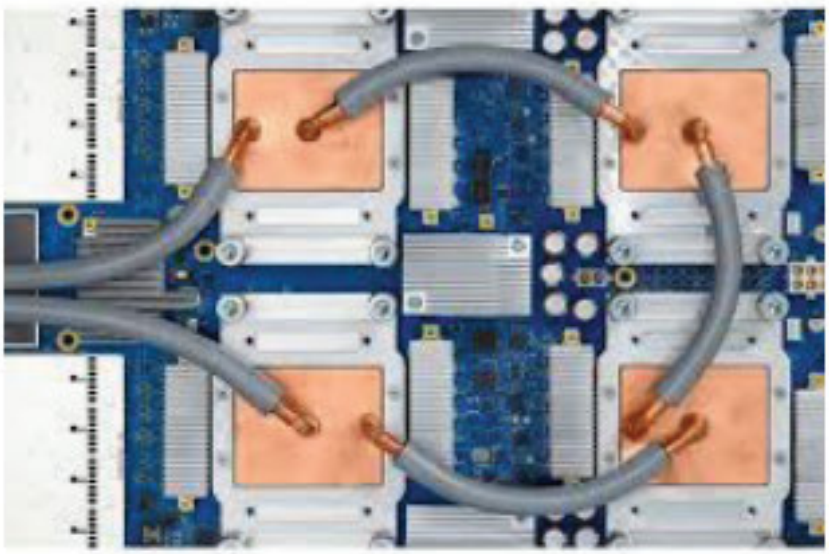
128. Google's TPUv2 and TPUv3 are coextensive with claims 1-8 of the '156 Patent. Set forth below is an analysis mapping the claims to the TPUv2 and TPUv3 devices.

A. Claim 1

1) "A device comprising:"

129. Each of the TPUv2 boards and TPUv3 boards are devices.

TPUv2 Board - Ex. 2016.

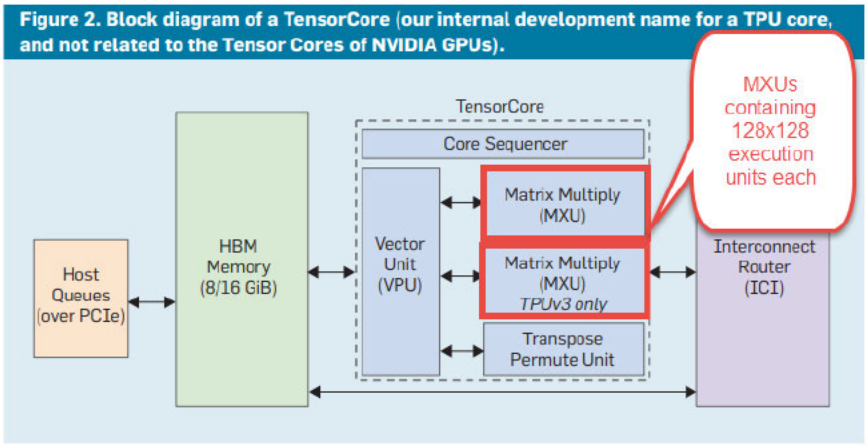


TPUv3 Board Ex. 2016.

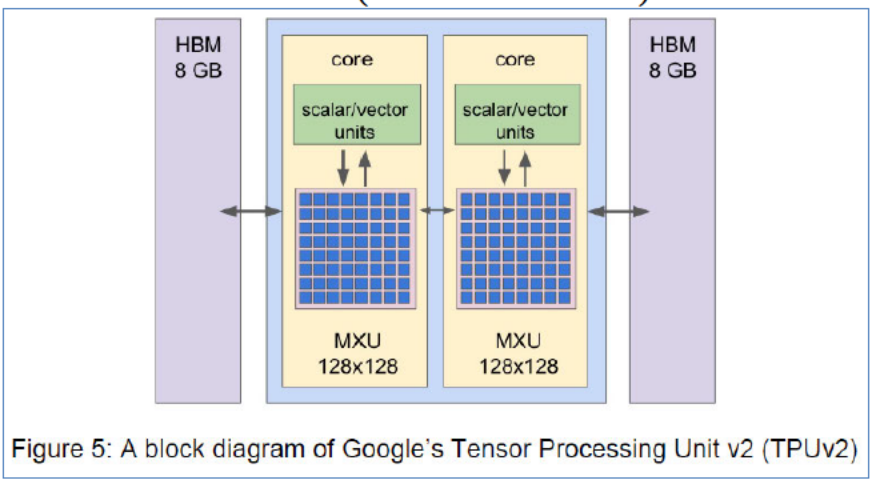
130. A POSA would understand that the above photographs show that the TPUv2 and TPUv3 boards are circuit boards coupled to hardware components such as processors, heatsinks, I/O ports, interconnected with each other via wiring. Both the TPUv2 and TPUv3 boards are "devices," as a POSA would understand that term.

2) "at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value"

131. In my opinion, each of the TPUv2 and TPUv3 boards include at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation (multiplication) on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value (the product of the first numerical value multiplied by another value). Both the TPUv2 and TPUv3 include at least one Matrix Multiply Unit (MXU) that contains 128 × 128 (*i.e.*, 16,384) execution units that each execute the operation of multiplication. Ex. 2016; Ex. 2011 ("the main computational capacity in each core provided by a large matrix multiply unit that can yield the results of multiplying a pair of 128 × 128 matrices each cycle").



Ex. 2016 (annotations added).



Ex. 2011.

132. Google's publications explain that each of the bfloat16 execution units in the MXU require "less circuitry" than either FP32 or FP16 multipliers:

> The IEEE fp32, IEEE fp16 and bfloat16 formats need 576 full adders, 121 full adders, and 64 full adders, respectively. Because multipliers for the bfloat16 format require so much less circuitry, it is possible to put more multipliers in the same chip area …

Ex. 2011, 9.

Each of the multipliers within the MXU is able to perform multiplication *only* at low-precision, using the bfloat16 format. They lack the circuitry to perform, for example, perform FP16 or FP32 operations. *Id. See also* Ex. 2049 at 3-4.

3) "wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/65,000 through 65,000 and"

133. As Google's publications explain:

> While its *inputs and outputs are 32-bit floating point values, the MXU typically performs multiplications at the reduced precision of bfloat16* — a 16-bit floating point representation that provides better training and model accuracy than the IEEE half-precision representation for deep learning as it allocates more bits to the exponent and less to the mantissa.

Ex. 2049 at 3-4 (emphasis added).

134. Thus, the dynamic range of the valid inputs to the multiplication operation in the TPUv2 and TPUv3 is governed by the number of exponent bits (8) in the floating point 32 (FP32) format. These inputs are supplied by the "scalar/vector units", also known as Vector Processing Unit (VPUs). As Google admits, 8 bits of exponent allow for a dynamic range of from roughly 2^{-126} (smaller than 1/65,000) through 2^{127} (larger than 65,000). Pet., 20-21; Ex. 1003, ¶ 232. I agree that the dynamic range of FP32 is at least this large. Indeed, in some of its publications, Google admits that floating-point formats with 8 bits of exponent are capable of representing an even wider range of values, from approximately 10^{-38} to 3×10^{38}. Ex. 2041 at 1.

4) "for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from

the result of an exact mathematical calculation of the first operation on the numerical values of that same input;"

135. Google's TPUv2 and TPUv3 boards meet this element. As I explain above, the MXU's inputs are "32-bit floating point" values, but they "perform[] multiplications at the reduced precision of bfloat16." *See* Ex. 2049 ("While its inputs and outputs are 32-bit floating point values, the MXU typically performs multiplications at the reduced precision of bfloat16").

136. Multiplication performed at bfloat16 precision uses 7 bits for the mantissa. Ex. 2041 at 1; Ex. 2011 at 8. Petitioner states – and I agree – that utilizing 7 bits of mantissa in multiplication operations results in a minimum of 12% of valid floating point 32 inputs, producing at least 0.39% relative error compared to the exact mathematical calculation of a full-precision multiplication on those same inputs. Pet., 73. Accordingly, the TPUv2 and TPUv3 meet this element.

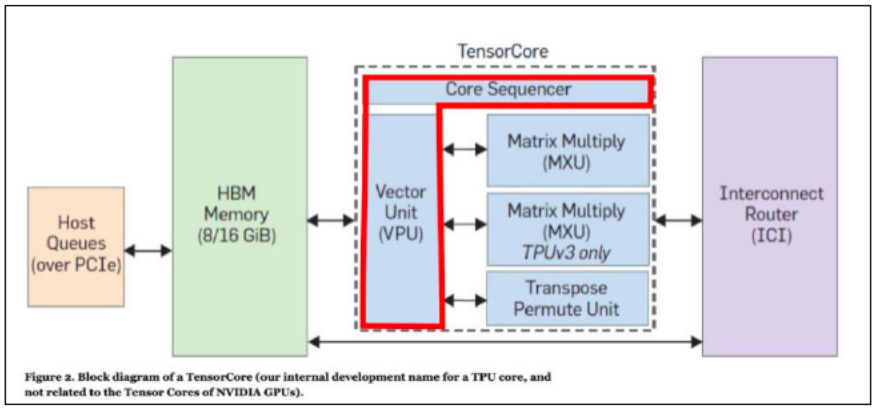
5) "at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit."

137. Google's TPUv2 and TPUv3 devices meet this limitation as well. Each TPU device is controlled by a Core Sequencer (which in turn is controlled by a Host Virtual Machine that is a program running on a CPU). The Core Sequencer is a a computing device, and by issuing VLIW instructions to the MXU, it controls the operation of the LPHDR execution units within the MXUs:

> The *Core Sequencer* fetches VLIW (Very Long Instruction Word) *instructions* from the core's on-chip, software-managed Instruction Memory … The 322-bit VLIW instruction can launch eight operations: two scalar, two vector ALU, vector load and store, and a pair of slots that *queue data to and from the matrix multiply and transpose units*.

Ex. 2016, 4-5.

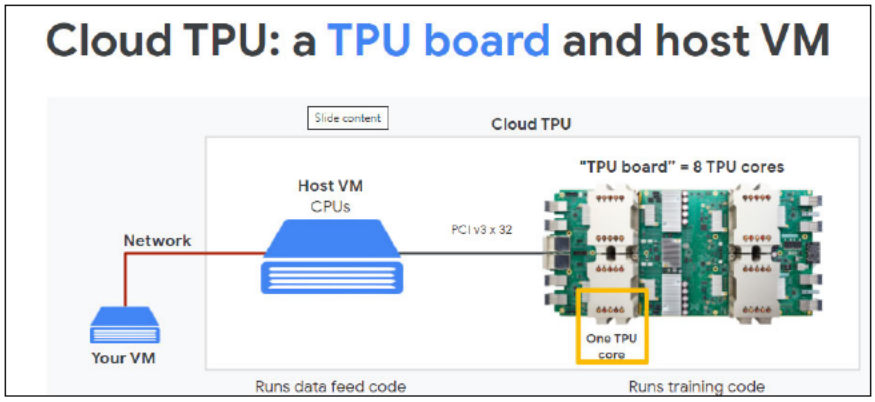
138. The CPU is a computing device, and by running the Host VM and issuing commands to the Core Seqeuncer, it controls the operation of the LPHDR execution units within the MXUs.



Ex. 2016 (annotations added).

Die sizes are adjusted by the square of the technology, as the semiconductor technology for TPUs is similar but larger and older than that of the GPU. We picked 15nm for TPUs based on the information in Table 3. Thermal Design Power (TDP) is for 16-chip systems. TPUs come with a host CPU. This GPU price adds price of a n1-standard-16 CPU.

Ex. 2011.



Ex. 2045, 2.

B. Claim 2

1) "The device of claim 1, wherein the at least one first computing device comprises at least one of a central processing unit (CPU), a graphics processing unit (GPU), a field programmable gate array (FPGA), a microcode-based processor, a hardware sequencer, and a state machine."

139. As discussed above, the Host VMs run on a CPU.

140. A person of ordinary skill in the art would also understand that the Core Sequencer described above is at least a "hardware sequencer" and a "state machine," based on the description of its function in Google's own publication:

> The *Core Sequencer* fetches VLIW (Very Long Instruction Word) *instructions* from the core's on-chip, software-managed Instruction Memory … The 322-bit VLIW instruction can launch eight operations: two scalar, two vector ALU, vector load and store, and a pair of slots that *queue data to and from the matrix multiply and transpose units*.

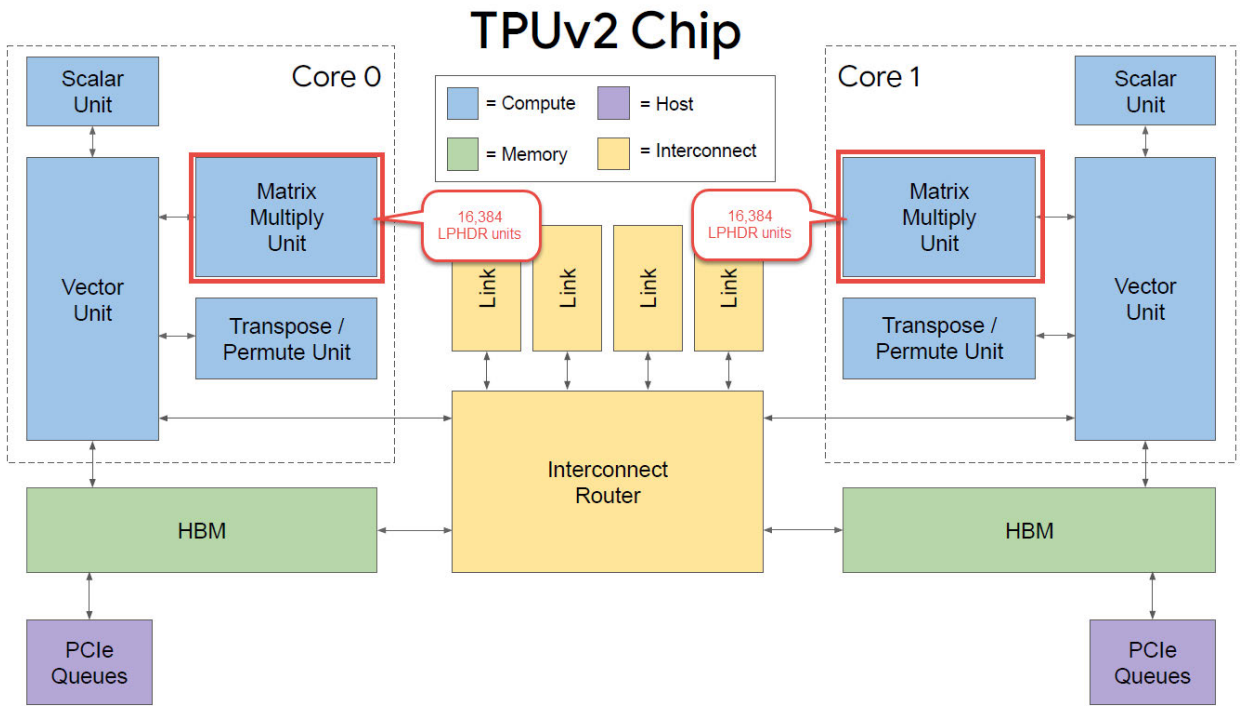
Ex. 2016, 4-5.

141. Therefore, the Core Sequencer and the CPU, independently and acting in concert, satisfy the additional requirement imposed by claim 2.

C. Claim 3

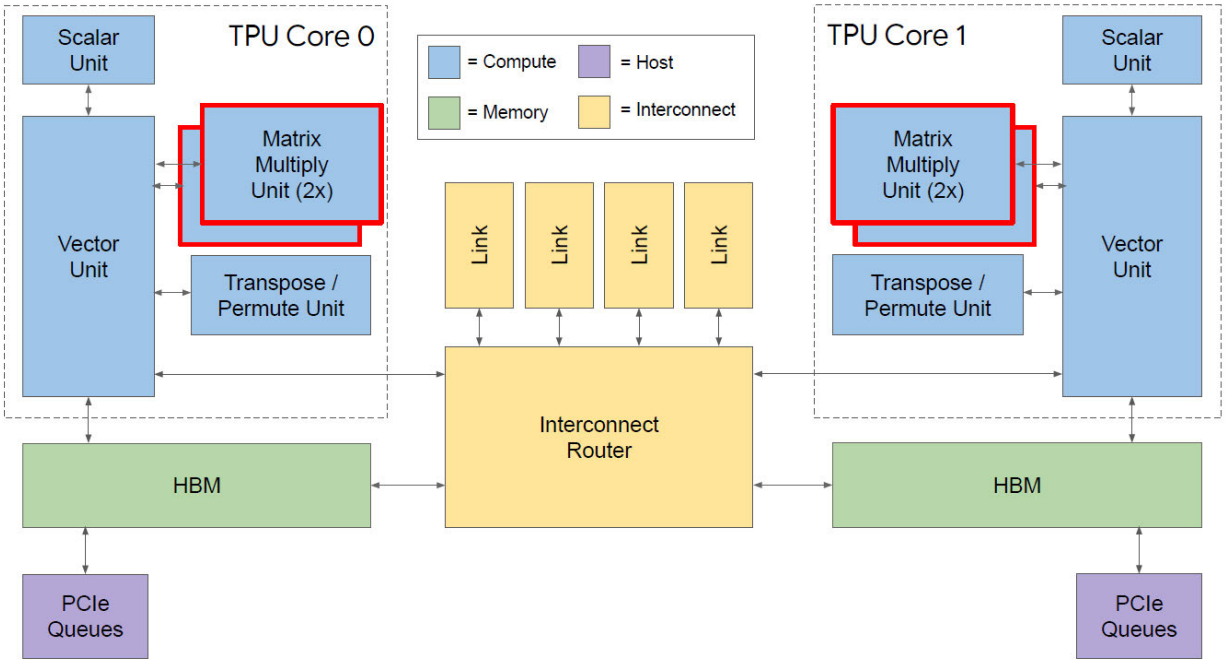
1) *"The device of claim 2, wherein the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide."*

142. Google's TPUv2 and TPUv3 boards meet this limitation. As discussed above, each MXU contains 128x128 execution units: a total of 16,384 LPHDR execution units in total per MXU. The TPUv2 board includes 1 MXU per core, with 2 cores in a chip, and 4 chips on a board. Ex. 2016, 6. Thus, the TPUv2 board contains 128 × 128 × 2 × 4 = 131,072 LPHDR execution units.



TPUv2 Board – Ex. 2046 (annotations added)

143. The TPUv3 board contains 2 MXUs per core, 2 cores per chip, and 4 chips per board. Ex. 2016. Thus, a single TPUv3 board contains 128 × 128 × 2 × 2 × 4 = 262,144 LPHDR execution units.



TPUv3 Board – EX. 2046

144. The MXU execution units perform multiplication at bfloat16 precision, and are not capable of performing full-precision multiplication, as I have discussed above.

145. Based on my review of Google's public documents, and the documents Google has produced in this IPR proceeding, each core (whether TPUv2 or TPUv3) contains a Vector Unit (VPU). *Id.* Each VPU contains 128 × 8 ×

2 32-bit arithmetic logic units (ALUs), for a total of 2048 ALUs per core. Ex. 2046, 34.

146. 32-bit ALUs are adapted to perform the operation of multiplication on 32-bit numbers. With two cores per chip and 4 chips per board, each TPUv2 or TPUv3 board contains 2048 × 4 × 8 = 16,384 execution units that are adapted to perform multiplication on 32-bit numbers.

147. As I explain above, each TPUv2 board contains 131,072 LPHDR execution units, and each TPUv3 board contains 262,144 LPHDR execution units. Therefore, in both the TPUv2 and TPUv3 boards, "the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide" and each board therefore meets the additional requirement imposed by claim 3.

D. Claim 4

1) The device of claim 3, wherein X=10%.

148. See Section IX.A.4 (mantissa of 7 bits gives at least 12% X).

E. Claim 5

1) The device of claim 3, wherein Y=0.2%

149. See Section IX.A.4 (mantissa of 7 bits gives at least 0.39% Y).

F. Claim 6

1) The device of claim 3, wherein X=10% and Y=0.2%

150. See Section IX.A.4 (mantissa of 7 bits gives at least 12% X and 0.39% Y).

G. Claim 7

1) The device of claim 3, wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000

151. As discussed above, the bfloat16 format used in the TPUv2 and TPUv3 boards contains 8 bits of exponent, which provides a dynamic range of from roughly 2^{-126} (smaller than 1/1,000,000) through 2^{127} (larger than 1,000,000).

H. Claim 8

1) The device of claim 3, wherein the first operation is multiplication

152. As discussed above, the MXUs within the TPUv2 and TPUv3 boards perform multiplication.

X. CONCLUSION

153. In signing this declaration, I recognize that the declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross-examination in the case and that cross-examination will take place within the United States. If cross-examination is required of me, I will appear for cross-examination within the United States during the time allotted for cross-examination.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Executed this 9th Day of August, 2021

Respectfully submitted,

Sunil P Khatri, Ph.D.